AMENDMENT IN THE CLAIMS:

Please amend claims 1, 6, and 11 as set forth in the complete claim listing below. This listing of claims will replace all prior versions and listings of claims in the application.

1. (Currently Amended) A method of providing a Certificate Status Service ("CSS") for checking validities of ~~authentication~~ certificates issued by respective issuing Certification Authorities ("CAs"), comprising the steps of:

receiving one or more certificate status queries from requesting entities;

if the issuing CAs are not found on a CSS's list of approved CAs or the certificates have expired, returning invalid statuses for those certificates;

if ~~the~~ current certificate statuses are found in ~~the CSS's status~~ a CSS cache memory, returning those certificates' statuses;

if any ~~status has~~ certificate statuses have not yet been determined, fetching, from a CSS configuration store, all certificate status reporting methods and communications information ~~from a configuration store of the CSS~~ that are needed for retrieving, from the respective issuing CAs, a certificate status of each certificate whose status has not yet been determined ~~from the respective issuing CAs~~;

configuring connectors based on the identified information for communicating with the issuing CAs;

communicating with the issuing CAs according to the configured connectors;

retrieving the ~~status~~ certificate statuses of all queried certificates;

processing the certificate statuses according to [[an]] the appropriate certificate status reporting ~~method~~ methods that may include, but is not limited to, Certificate Revocation Lists (CRLs) that are retrieved at specified publication intervals[[,]] and Delta Certificate Revocation Lists (ΔCRLs) that are retrieved upon notification, and LDAP, OCSP, and any other certificate status means that retrieve certificate statuses in real-time;~~are queried and retrieved using real-time protocols;~~

recording retrieved certificate statuses in the ~~CSS's~~ CSS cache memory;

returning the retrieved certificate statuses to the requesting entities;

wherein the issuing CAs and connector parameters, which enable the CSS to interwork with any CAs and CA domains even though the CSS and issuing CAs may operate using dissimilar certificate practices and policies, are designated on a list of approved CAs in ~~a~~ the CSS configuration store ~~that enable the CSS to interwork with any CAs and CA domains even though they can operate using dissimilar certificate practices and policies~~.

2. (Currently Amended) The method of claim 1, wherein a certificate indicating a validity period is deemed to have expired if a local date and time fall outside [[a]] the validity period ~~as indicated in the certificate~~.

3. (Currently Amended) The method of claim 2, wherein the issuing CA is added to at least one organization's list of approved CAs by vetting and approving the issuing CA according to predetermined business rules, wherein the business rules include at least one rule for reviewing the acceptability of the CA's certificate policy and

practices for insuring the identity of the entity requesting the certificate, and if the issuing CA is vetted and not approved or later disapproved, the issuing CA is added to the at least one organization's list of not-approved CAs in the CSS configuration store and/or has any prior entry removed from the at least one organization's list of approved CAs.

4. (Currently Amended) The method of claim 3, wherein vetting and approving the issuing CA include registering a representation of a trusted ~~authentication~~ certificate of the CA with the CSS and adding, to the CSS configuration store, at least [[a]] the certificate status reporting component of the CA[[,]]; the certificate status reporting method including, but not limited to CRL, OCSP, or LDAP[[,]]; a time-to-live data element[[,]]; and communication information needed to configure a connector ~~to the CSS's configuration store~~.

5. (Currently Amended) The method of claim 4, further comprising the steps of:

checking and updating ~~a local~~ the CSS cache memory for the ~~certificate~~ queried certificate status, and if the queried certificate status is found in the ~~local~~ CSS cache memory, checking that the local date and time are within the certificate's validity period and that the time-to-live data element and use-counter values are within a threshold;

if any of the validity period, time-to-live data element, or use-counter values are unacceptable, clearing the ~~local~~ CSS cache memory, wherein if the certificate status is not found in the ~~local~~ CSS cache memory, the CSS establishes a communication session with the certificate status reporting component of the issuing CA, composes a certificate

status request using one of the CRL or real-time reporting methods according to the configured connector, retrieves the certificate status from the certificate status reporting component, closes the communication session with the certificate status reporting component, and adds at least one of the certificate's identification, certificate's status, use-counter, and time-to-live data element to the ~~local~~ CSS cache memory.

6. (Currently Amended) The method of claim 1, wherein the certificate status reporting method is indicated to be a Certificate Revocation List, according to a publication schedule of the issuing CA, wherein the CSS retrieves the CRL from a certificate status reporting component listed in the CSS configuration store, the CSS clears the CSS cache memory associated with the issuing CA, and the CSS extracts the certificate ~~status~~ statuses of all ~~authentication~~ certificates from the CRL and stores the certificate statuses in the CSS cache memory associated with the issuing CA.

7. (Currently Amended) The method of claim 1, wherein the certificate status reporting method is indicated to be a ΔCRL, wherein upon notification by the issuing CA that the ΔCRL is available, the CSS retrieves the ΔCRL from a certificate status reporting component listed in the CSS configuration store and if the ΔCRL is a full CRL, then the CSS clears the CSS cache memory associated with the issuing CA, extracts all certificate statuses from the CRL, and stores the certificate statuses in the CSS cache memory, and if the ΔCRL contains changes occurring after publication of a full CRL, the CSS extracts all certificate statuses from the ΔCRL, and stores the certificate statuses in the CSS cache memory.

8. (Previously Presented) The method of claim 1, wherein the communicating step includes communicating according to a plurality of connectors to multiple CAs and PKIs.

9. (Currently Amended) The method of claim 1, wherein the ~~connector allows~~ connectors allow more than one certificate status request to be chained together in a single communicating step between the CSS and the issuing CA.

10. (Currently Amended) The method of claim 1, wherein the certificates are held in the CSS configuration store until expiration and information ~~are~~ is extracted as needed.

11. (Currently Amended) The method of claim 1, wherein the retrieving of the ~~status~~ statuses of the ~~certificate~~ certificates issued by the approved [[CA]] CAs in response to ~~a query~~ queries from a trusted third-party repository of information objects to the CSS to validate the ~~authentication certificate's status~~ certificate statuses comprises the steps of:

locating and reporting the ~~status~~ certificate statuses if the ~~status is~~ certificate statuses are present and current in the CSS cache memory; ~~of the CSS;~~

if the ~~status is~~ certificate statuses are not present in the CSS cache memory, performing the steps of:

obtaining the communications information, certificate status ~~type~~ types, and retrieval ~~method~~ methods from the CSS configuration store;

if the certificate status type is CRL, ~~and~~ the CRL in the CSS cache memory is current, and the certificate statuses ~~is~~ are not found in the CSS cache memory, then reporting the certificate statuses as valid; and

if the certificate status type is CRL, the CRL is not current or found in the CSS cache memory, and local time is greater than a next scheduled publication time for the CRL, or if the certificate status type is not CRL,

creating [[a]] connectors and composing [[a]] certificate status requests according to the certificate status type;

establishing [[a]] communication sessions with [[a]] the certificate status reporting components of the issuing CAs;

retrieving the certificate statuses from the ~~CA's~~ certificate status reporting components using the obtained retrieval methods and ending the communication sessions;

interpreting the retrieved certificate statuses;

associating, with the interpreted retrieved certificate statuses, [[a]] time-to-live values representing [[a]] periods specified by the respective CSS ~~policy~~ policies for the certificate status types;

adding at least one of the certificate's identification, certificate status and time-to-live values to the CSS cache memory; and

reporting the certificate statuses to the trusted third-party repository of information objects.

12. (Canceled)

13. (Canceled)

14. (Canceled)

15. (Currently Amended) The method of claim 1 for providing certificate status reports for ~~authentication~~ certificates issued by the approved CAs further comprising:

reporting valid certificate ~~status~~ statuses when the certificate status type is CRL, the CRL is current, ~~or~~ and the certificate statuses ~~is~~ are not found in the CSS cache memory;

reporting the certificate statuses when the certificate statuses ~~is~~ are found in the CSS cache memory and the time-to-live and use-counter values have not exceeded respective thresholds; otherwise,

if either the time-to-live or use-counter values have exceeded ~~the threshold~~ respective thresholds clearing the ~~status~~ certificate statuses from the CSS cache memory;

if the certificate statuses ~~has~~ have not been reported in a previous step, then requesting and retrieving the certificate statuses using the certificate status type indicated in the CSS configuration store;

when the status type is CRL, retrieving and parsing the new CRL at a next indicated publication time;

when the certificate status type is at least one of the type LDAP, OSCP, and any other real-time certificate status reporting protocol, retrieving and parsing the certificate status;

adding at least one of the certificate's identification, certificate status, time-to-live and use-counter values to the CSS cache memory; and

reporting the retrieved ~~status~~ certificate statuses to the requesting entity.

16. (Currently Amended) The ~~CSS~~ method of claim 15, wherein a certificate status use-counter data element is added to the CSS's certificate status cache ~~memory~~, wherein the certificate status use-counter data element is incremented or decremented every time the certificate's status is checked, and if the certificate status use-counter ~~data element passes a~~ value exceeds respective threshold, then the certificate status is reported and the CSS cache memory is cleared with respect to the certificate status.

17. (Currently Amended) The ~~CSS~~ method of claim 16, wherein a certificate status last-accessed data element is added to the CSS cache memory, and the certificate status last-accessed data element in conjunction with the certificate status use-counter data element enable the CSS to determine an activity level of the certificate's status.

18. (Currently Amended) The ~~CSS~~ method of claim 17, wherein when a request is made to the CSS to retrieve a certificate status of a new certificate and the CSS cache memory has reached an allocated ~~buffer~~ memory size limit, the CSS searches the CSS cache memory for every certificate status entry where the current time exceeds the time-to-live value for every certificate status entry where the value of the use-counter data element exceeds the threshold and the value of the at least one certificate status entry with the oldest last-accessed value, wherein the CSS then clears the respective CSS cache

memory entries, retrieves the requested certificate status, places the certificate status in the CSS cache memory, and reports the requested certificate status to the requesting entity.

19. (Withdrawn) A method of executing a transaction between a first party and a second party by transferring control of an authenticated information object having a verifiable evidence trail, comprising the steps of:

retrieving an authenticated information object from a trusted third-party repository of information objects, wherein the authenticated information object includes a first digital signature block comprising a digital signature of a submitting party and a first authentication certificate relating at least an identity and a cryptographic key to the submitting party, a date and time indicator, and a second digital signature block comprising a second digital signature of the trusted third-party repository of information objects and a second authentication certificate relating at least an identity and a cryptographic key to the trusted third-party repository of information objects; the first digital signature block was validated by the trusted third-party repository of information objects; and the authenticated information object is stored as an authoritative copy information object under the control of the trusted third-:-party repository of information objects;

executing the retrieved authenticated information object by the second party by including in the retrieved authenticated information object a third digital signature block comprising at least a third digital signature and a third authentication certificate of the second party; and

forwarding the executed retrieved authenticated information object to a trusted third-party repository of information objects, wherein the trusted third-party repository of information objects verifies digital signatures and validates authentication certificates associated with the digital signatures included in information objects by at least retrieving status of the authentication certificates from a Certificate Status Service ("CSS") provided according to claim 1; the trusted third-party repository of information objects rejects a digital signature block if the respective digital signature is not verified or the status of the respective authentication certificate is expired or is revoked; and if at least one signature block in the information object is not rejected, the trusted third-party repository of information objects appends the trusted third-party repository's digital signature block and a date and time indicator to the information object and takes control of the object on behalf of the first party.

20. (Withdrawn) The method of claim 19, wherein a signature block includes at least one hash of at least a portion of the information object in which the signature block is included, the at least one hash is encrypted by the cryptographic key of the block's respective signer, thereby forming the signer's digital signature, and the signer's digital signature is included in the signature block with the signer's authentication certificate.

21. (Withdrawn) The method of claim 20, wherein the executing step includes displaying a local date and time to the second party, affirming, by the second party, that the displayed local date and time are correct, and correcting the local date and time if either is incorrect.

22. (Withdrawn) The method of claim 19, wherein if the trusted third-party repository of information objects rejects a digital signature block, the trusted third-party repository of information objects requests a remedy that requires the digital signature to be recomputed and the signature block to be reforwarded.

23. (Withdrawn) The method of claim 19, wherein the trusted third-party repository of information objects checks the local date and time for accuracy and that they are within a validity period indicated by the second party's authentication certificate.

24. (Withdrawn) The method of claim 23, wherein if the local date and time are not within the validity period indicated by the second party's authentication certificate, the trusted third-party repository of information objects notifies the second party that the authentication certificate is rejected and the first party that the transaction is incomplete.

25. (Withdrawn) The method of claim 19, wherein one or more digitized handwritten signatures are included in the information object, and placement of the digitized handwritten signatures in a data structure is specified by at least one signature tag.

26. (Withdrawn) The method of claim 19, wherein placement of one or more signature blocks in a data structure is specified by at least one signature tag.

27. (Withdrawn) The method of claim 26, wherein one or more signature blocks are separately forwarded to the trusted third-party repository of information objects with respective signature tags, and the trusted third-party repository of information objects validates the signature blocks by:

rejecting a signature block if either the respective digital signature is not verified or the respective authentication certificate is not validated, and

placing the signature block according to the respective signature tag if the signature block is not rejected,

wherein, to signature blocks sent separately, the trusted third-party repository of information objects adds a date and time indication to each signature block and appends according to business rules the trusted third-party repository's signature block in a wrapper that encompasses the information object and placed signature blocks.

28. (Withdrawn) The method of claim 27, wherein the trusted third-party repository of information objects verifies a digital signature and validates an authentication certificate in a signature block by:

determining from the business rules whether a party associated with the authentication certificate has authority, verifying the party's digital signature, checking that the authentication certificate's validity period overlaps the trusted third-party repository's current date and time,

checking that the local date and time falls within an allowable deviation from the trusted third-party repository's current date and time, and

retrieving status of the authentication certificate from the CSS, and

if any of the preceding steps results in an invalid or false output, the digital signature is deemed invalid, the transaction is not executed, otherwise the digital signature is deemed valid and the transaction is executed.

29. (Withdrawn) The method of claim 19, wherein the CSS provides authentication certificate status to the trusted third-party repository of information objects by at least the steps of checking a local cache memory for the status, and if the status is found in the local cache memory and the local date and time are within the validity period, and retrieving the status from the local cache memory; or if the time-to-live or use-counter threshold is exceeded clearing the cache memory entry, wherein if the status is not found in the local cache memory, the CSS establishes a communication session with a certificate status reporting component of the issuing CA, composes a certificate status request, retrieves the status from the certificate status reporting component. closes the communication session with certificate status reporting component, and adds at least the authentication certificate's identification, status, and a time-to-live data element to the local cache memory.

30. (Withdrawn) The method of claim 19, wherein the first party is a first trusted third-party repository of information objects and the transaction is for transferring custody of one or more authoritative copies to the first trusted third-party repository of information objects from a second trusted third-party repository of information objects, an owner of the transaction provides the second trusted third-party repository of information objects with a manifest that identifies authoritative copies to be transferred to

the first trusted third-party repository of information objects, the second trusted third-party repository of information objects establishes communication with the first trusted third-party repository of information objects and identifies the purpose of its actions, the manifest is communicated to the first trusted third-party repository of information objects so that it is able to determine when the transfer of custody has been completed, the second trusted third-party repository of information objects transfers each identified authoritative copies to the first trusted third-party repository of information objects, the first trusted third-party repository of information objects retrieves status of the second trusted third-party repository's certificate and verifies the second trusted third-party repository's digital signature on each transferred authoritative copies, if any of the second trusted third-party repository's digital signatures or certificates are invalid, then the first trusted third-party repository of information objects notifies the second rusted third-party repository of information objects and seeks a remedy I if the second trusted third-party repository of information objects does not provide a remedy, the first trusted third-party repository of information objects notifies the transaction owner that the requested transfer of custody has failed, otherwise the second trusted third-party repository of information objects creates a new wrapper for each successfully transferred information object, adding a date-time stamp and the first trusted third-party repository's signature block.

31. (Withdrawn) The method of claim 30, wherein the transaction is a transfer of ownership in response to an instruction, transfer of ownership documentation is placed in either the first trusted third-party repository of information objects or the second trusted third-party repository of information objects, the trusted third-party repository of

information objects having the transfer of ownership documentation validates authenticity of the transfer of ownership documentation by verifying all digital signatures, certificate validity periods, and using the CSS to check certificate status of all authentication certificates included in the transfer of ownership documentation, appends a date and time indication, and digitally signs, wraps and stores the transfer of ownership documentation, which are added to the manifest.

32. (Withdrawn) The method of claim 19, wherein certificate status is indicated to the CSS by a Certificate Revocation List ("CRL"), according to a publication schedule of the issuing CA, the CSS retrieves the CRL from a certificate status reporting component listed in the configuration store, the CSS clears a cache memory associated with the issuing CA, and the CSS determines the status of the authentication certificate from the CRL and stores the status in the cache memory associated with the issuing CA.

33. (Withdrawn) The method of claim 19, wherein certificate status is indicated to the CSS by a Delta Certificate Revocation List ("ΔCRL"); upon notification by the issuing CA that a ΔCRL is available, the CSS retrieves the ΔCRL from a certificate status reporting component listed in the configuration store; if the ΔCRL is a complete CRL, then the CSS clears a cache memory associated with the issuing CA, determines the status from the CRL, and stores the status in the cache memory; and if the ΔCRL contains only changes occurring after publication of a full CRL, the CSS determines the status from the ΔCRL, and stores the status in the cache memory.

34. (Previously Presented) The method of claim 18, wherein a cleanup process removes all stale cache entries as required when new CRLs or ΔCRLs are retrieved, one of the thresholds is exceeded, or freeing up of cache is required.

35. (Currently Amended) The method of claim 1, wherein any CSS can query any other CSS for the certificate status if that other CSS is designated in the CSS configuration store as an approved certificate status reporting component for the issuing CA.